

Does Understanding Inform Generation in Unified Multimodal Models? From Analysis to Path Forward

Yuwei Niu^{1,3,*}, Weiyang Jin^{4,*}, Jiaqi Liao⁵, Chaoran Feng¹, Peng Jin⁶,
Bin Lin¹, Zongjian Li¹, Bin Zhu¹, Weihao Yu^{1,2}, Li Yuan^{1,2,†}

¹Shenzhen Graduate School, Peking University ²PengCheng Laboratory

³Chongqing University ⁴The University of Hong Kong

⁵The Hong Kong University of Science and Technology ⁶Qwen Team

*Equal contribution. †Corresponding author.

niuyuwei04@gmail.com, yuanli-ece@pku.edu.cn

Abstract

Recent years have witnessed significant progress in Unified Multimodal Models, yet a fundamental question remains: Does understanding truly inform generation in Unified Multimodal Models? To investigate this, we introduce **UniSandbox**, a decoupled evaluation framework paired with controlled, synthetic datasets to avoid data leakage and enable detailed analysis. Our findings reveal a significant understanding-generation gap, which is mainly reflected in two key dimensions: reasoning generation and knowledge transfer. Specifically, for reasoning generation tasks, we observe that explicit Chain-of-Thought (CoT) in the understanding module effectively bridges the gap, and further demonstrate that a self-training approach can successfully internalize this ability, enabling implicit reasoning during generation. Additionally, for knowledge transfer tasks, we find that CoT assists the generative process by helping retrieve newly learned knowledge, and also discover that query-based architectures inherently exhibit latent CoT-like properties that affect this transfer. UniSandbox provides preliminary insights for designing future unified architectures and training strategies that truly bridge the understanding-generation gap.

1 Introduction

Recent years have witnessed significant progress in Unified Multimodal Understanding and Generation Models (Wu et al., 2024c,a; Dong et al., 2024; Liao et al., 2025b; Fang et al., 2024; Xie et al., 2024; Lin et al., 2025; Li et al., 2025e,d; Wu et al., 2025c), with powerful architectures designed to integrate both tasks. Yet this trend raises a more fundamental question: what is the true synergy gained by housing both capabilities in a single model? We investigate one crucial direction: whether a model’s rich language priors can genuinely inform and guide its visual generation.

Existing works (Tong et al., 2024; Sun et al., 2025; Xie et al., 2025c; Zou et al., 2025a; Li et al.,

2025c), such as WISE (Niu et al., 2025), aim to evaluate whether models can use world knowledge for visual generation in complex contexts. For instance, given the prompt “A flower commonly gifted on Mother’s Day,” an ideal model would first deduce “carnation” from its world knowledge, and this concept would then serve as a clear instruction for the visual generator, demonstrating a true synergy in which understanding guides generation.

However, while these works valuably reveal that models perform poorly when required to use internal knowledge or reasoning for generation, they conflate multiple failure modes and offer no fine-grained attribution. We cannot tell whether a failure stems from a *knowledge* deficit (the model lacks the facts), a *reasoning* deficit (it cannot deduce the answer from known facts and rules), or a *transfer* failure (the understanding module knows the answer, but the generation module cannot use it). Moreover, potential data leakage renders the results unreliable, as a model’s “success” might stem from memorizing training pairs rather than genuine reasoning. This dual problem of non-attributability and unreliability makes it hard to identify which architectures or training strategies best foster understanding-generation synergy.

To analyze the understanding-generation gap and address the above issues, we introduce **UniSandbox**¹, a decoupled framework with synthetic, low-leakage data. UniSandbox isolates understanding into two dimensions, *Knowledge* and *Reasoning*, allowing us to precisely attribute model performance beyond data memorization, and we use it to evaluate several state-of-the-art unified models.

For reasoning generation, we design reasoning-driven tasks that require generating images from mathematical calculation or logical deduction. Without explicit Chain-of-Thought (CoT), all open-

¹Code and data are available at <https://github.com/PKU-YuanGroup/UniSandBox>.

source models score near zero, indicating that their generation degenerates into a shallow word-to-pixel mapping, essentially a “bag-of-words” (Yuksekgonul et al., 2022). With CoT, the understanding module solves the problem and explicitly guides generation. We further explore whether a simple self-training method can internalize this capability without explicit CoT, turning explicit reasoning implicit and offering a direction for optimizing training paradigms.

In parallel, for knowledge transfer, we inject new knowledge into the understanding module and test whether the generation module can reproduce it through image generation. Mainstream architectures fail this task, while CoT enables internal knowledge retrieval that transfers the knowledge to the generation module, substantially improving performance. Visualization analysis further reveals that query-based architectures, which use extra queries to extract information as the generation condition, exhibit a latent CoT-like capability.

The contributions of our work are as follows:

- **The UniSandbox Framework.** We propose UniSandbox, a novel decoupled evaluation framework for investigating whether understanding truly informs generation. Through synthetic data and fine-grained analysis, our framework reveals a significant understanding-generation gap in existing models.
- **Revealing Flaws in Reasoning Generation.** We confirm the severe deficiency of existing models in reasoning-based generation, demonstrate the role of Chain-of-Thought as an explicit “bridge”, and explore how this capability can be internalized through self-training.
- **Investigating the Knowledge Transfer Bottleneck.** We confirm that models struggle to transfer newly acquired knowledge to the generation module, and again identify CoT as an effective activator. Furthermore, we find that query-based architectures exhibit a “latent knowledge transfer capability”, providing a key design reference for future UMMs.

2 UniSandbox: Controlled Evaluation

To investigate whether and how the “understanding” capabilities of Unified Multimodal Models benefit their “generation” tasks, we decompose “understanding” into two core dimensions: *Knowledge* (the memory and retrieval of factual information)

and *Reasoning* (the application of procedural rules). This enables fine-grained attribution, pinpointing whether a model’s failure stems from an inability to invoke newly acquired knowledge or to apply logical reasoning when guiding image generation. To reduce the risk of data contamination from overlap between pre-training and evaluation data, we build our training and evaluation pipelines on controlled synthetic data designed to minimize such overlap, providing a “sandbox environment” for analysis.

We selected a representative set of models spanning three paradigms: (1) Autoregressive (AR), e.g., Janus-Pro-7B (Chen et al., 2025d); (2) AR + Diffusion (shallow fusion), including Qwen-Image (Wu et al., 2025a), which uses the last hidden state as the generation condition, and Blip3o (Chen et al., 2025b), which uses queries to extract conditional information; and (3) AR + Diffusion (deep fusion), e.g., BAGEL (Deng et al., 2025), which deeply integrates understanding and generation in one transformer.

3 Reasoning Generation

We first focus on the “reasoning” dimension to explore whether the model’s internal logic can guide generation, designing two families of synthetic tasks: **Mathematical Operations** and **Symbolic Mapping**. These serve as ideal probes for core reasoning: both readily produce out-of-distribution data, their difficulty scales precisely with the number of computational steps or the mapping depth, and they compel the model to move beyond superficial semantics to abstract, procedural symbol manipulation. This lets us quantify the extent to which the generative module inherits the reasoning capabilities of its understanding module. All task instances were generated using GPT-4o, with detailed prompts in Appendix A.

3.1 Task Design: Explore Reasoning Capabilities

3.1.1 Mathematical Operations

Unlike previous benchmarks (Ghosh et al., 2023; Huang et al., 2023) that test a model’s basic generative capacity by directly specifying the number of objects, we require the model to perform sequential calculations and then generate a corresponding number of objects. For instance, with a prompt like “Provide the same number of erasers as calculated by $3 - 2$,” the model must first understand and execute the $3 - 2$ operation to derive the answer 1, and

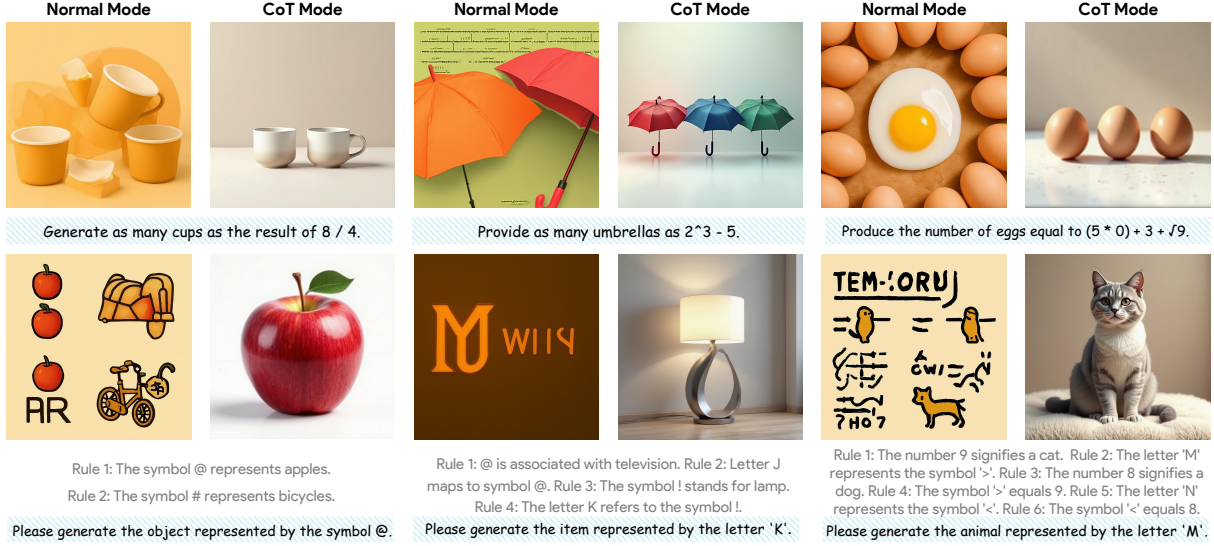


Figure 1: Data examples for reasoning generation. All images are generated by BAGEL. Normal and CoT represent generation without and with think mode (Chain-of-Thought mode), respectively. We also show the corresponding prompts.

then use that internally-reasoned result to guide the visual generation of one eraser. This design compels the model to engage its internal reasoning and comprehension abilities before executing the generation task. The difficulty of our tasks progressively increases by extending the length of the operation chain. We use a variety of basic middle-school-level operations, including addition, subtraction, multiplication, division, exponentiation, and modulo operations. All final results are constrained to be integers less than seven, ensuring that the model’s potential inability to generate a large number of objects does not interfere with the evaluation. The tasks are structured across three levels of difficulty: first-level tasks require a single arithmetic calculation, while higher-level tasks demand sequential deductions involving two or three distinct, composite operations. Each of the three levels contains 100 prompts, for a total of 300; see Figure 1 for examples.

3.1.2 Symbolic Mapping

The symbolic mapping tasks are designed to evaluate a model’s ability to follow novel, arbitrary rules by reasoning through mapping chains of varying lengths. Unlike a simple text-to-image task that requires direct mapping (e.g., “apple” to the image of the apple), our method compels the model to reason through a set of given mapping rules, requiring it to identify and generate a unique target object from two given options. For instance, a task might be defined by a set of arbitrary rules: “The

letter ‘A’ represents the number 1, and the number 1 represents a cat.” Instead of directly asking for a cat, we prompt the model to generate the animal represented by the letter ‘A.’ To succeed, the model must autonomously reason through a two-step chain—from ‘A’ to ‘1,’ and then from ‘1’ to ‘cat’—before it can generate the correct image. This design compels the model to leverage its internal reasoning to bridge the gap between abstract symbols and a concrete visual output. We progressively increase the difficulty by extending the length of the mapping chain, from a one-step mapping in first-level tasks to two or three-step chains in higher-level ones. To prevent models from succeeding by random guessing, we employ paired prompts, where each pair is based on the same set of rules but asks for a different result. We generated 100 prompt pairs for each of the three difficulty levels, for a total of 600 individual prompts; see Figure 1 for examples.

3.2 Evaluation Protocol

To avoid the judge biases of using Multimodal Large Language Models (MLLMs) as direct visual judges (Chen et al., 2025a), we use a two-stage protocol: an MLLM first captions each generated image, and then compares the caption with the ground-truth answer without access to the image, awarding a score of 1 if consistent and 0 otherwise. For symbolic mapping, we apply a stricter criterion: a trial is successful only if the model correctly generates the output for **both** prompts in a given pair,

ensuring that success is attributable to genuine reasoning rather than random guessing. Evaluation details are in Appendix C.

3.3 From Reasoning Failure to CoT Activation

The results, presented in Table 1, reveal a pervasive deficiency. Most models, particularly the open-source ones, show severe limitations on the mathematical operation and symbolic mapping tasks that require logical reasoning: their performance is consistently near zero, and on the more abstract symbolic mapping tasks they fail entirely. This supports our central thesis that the generative component of current unified models operates as a shallow keyword-mapping system. While capable of mapping simple keywords (e.g., generating an apple from the word “apple”), their generation breaks down when internal, procedural logic is required (e.g., first computing “3+2” and then generating “5 apples”). This indicates that the model’s internal comprehension is not effectively translated into generative action when logical reasoning is demanded.

However, a key comparative result points to a path toward bridging this gap. When Chain-of-Thought (CoT) is applied to BAGEL, its average score rises substantially, from 0.0283 (BAGEL) to 0.5083 (BAGEL + CoT). This provides strong evidence that **CoT serves as a crucial mechanism for activating the model’s latent language priors and guiding them into the visual generation process**. CoT compels the model to first articulate the reasoning steps and derive a conclusion in the language space, then use this explicit outcome as the generative instruction, effectively transforming a complex logical problem into a straightforward text-to-image task.

3.4 Can Reasoning Capabilities Be Internalized?

The above experiments demonstrate that CoT is a key mechanism for activating a model’s reasoning capabilities and applying them to generative tasks. This finding naturally raises a central question: can the explicit reasoning processes exhibited in CoT mode be **internalized** into the model’s standard, non-CoT generation process through self-training, thereby enabling it to fundamentally overcome its reasoning generation limitations?

Our framework builds on two key observations.

- **CoT as a “Teacher”:** As shown in the preceding section, CoT effectively guides the unified model (U_{CoT}) to perform correct logical reasoning, yielding high-quality reasoning-generation pairs that serve as a reliable source of “teacher” signals for fine-tuning.
- **Inherent Self-Verification Capability:** The unified model’s powerful visual understanding endows it with an inherent capability to evaluate its own generative output. This allows the model to accurately assess the semantic consistency between a generated image and a textual instruction. In other words, the understanding module (U_{Ver}) can serve as its own “verifier” to determine if its output faithfully adheres to the prompt, as has been demonstrated by prior works (Xie et al., 2025a; Jin et al., 2025; Mao et al., 2025).

Motivated by these insights, we propose a preliminary framework named **Self-Training with Rejection Sampling (STARS)**. Rather than a mature solution, it is part of the UniSandbox reasoning evaluation, exploring whether models can internalize explicit reasoning to aid their implicit reasoning during generation.

1. **Generation:** First, we leverage the model in CoT mode to generate training samples, D_{gen} , for a set of input instructions I from various reasoning tasks (e.g., mathematical operations, symbolic mapping). Each sample in this dataset contains the instruction, the corresponding CoT reasoning trace C , and the final output image O .

$$D_{gen} = \{(I_i, C_i, O_i) | (C_i, O_i) = U_{CoT}(I_i)\} \quad (1)$$

Each tuple (I_i, C_i, O_i) thus represents a complete instruction-thought-output chain.

2. **Filtering:** Next, we employ the model’s own understanding faculty, U_{Ver} , as a discriminator to filter D_{gen} . We define a **binary consistency score** $S(I, O) = U_{Ver}(I, O) \in \{0, 1\}$, where 1 signifies that the output O is consistent with the instruction I . Only samples with a score of 1 are retained. This process yields a curated, high-quality dataset, D_{fil} .

$$D_{fil} = \{(I_i, O_i) | (I_i, C_i, O_i) \in D_{gen} \wedge S(I_i, O_i) = 1\} \quad (2)$$

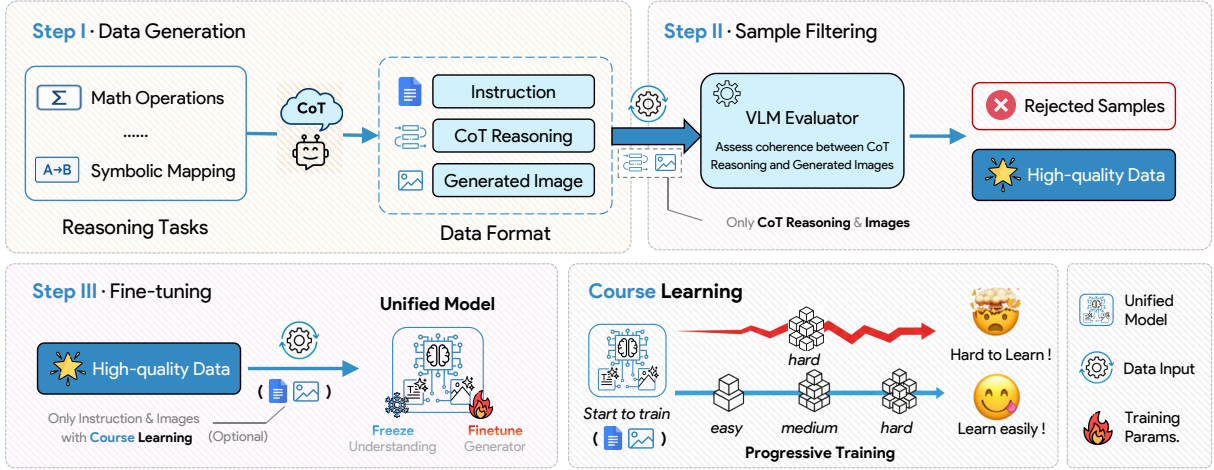


Figure 2: **Overview of the STARS framework.** It illustrates the three sequential stages: (I) Data Generation, where CoT is leveraged to create reasoning-generation pairs; (II) Sample Filtering, which uses the understanding module of UMMs to curate high-quality data; and (III) Fine-tuning, where the unified model is trained with the filtered data to distill CoT reasoning into its standard generation process.

Table 1: Comparison of model performance on the Math and Mapping tasks. Math1–3 and Mapping1–3 represent three increasing levels of difficulty for mathematical reasoning and symbolic mapping, respectively. **Bold values** indicate the best performance in each column.

Model	Math1	Math2	Math3	Mapping1	Mapping2	Mapping3	Average
Janus-Pro-7B	0.04	0.03	0.03	0.00	0.00	0.00	0.0167
Blip3o	0.03	0.02	0.02	0.03	0.00	0.00	0.0167
Qwen-Image	0.13	0.07	0.10	0.00	0.00	0.00	0.0500
BAGEL	0.07	0.06	0.04	0.00	0.00	0.00	0.0283
BAGEL + CoT	0.60	0.44	0.23	0.75	0.57	0.46	0.5083

Notably, the intermediate reasoning trace C is intentionally discarded at this stage, keeping only the (I, O) pairs.

3. **Fine-tuning:** Finally, the filtered dataset D_{fil} is used to fine-tune the original model’s standard generator, U_{Gen} , whose parameters are denoted by θ . The objective is to train the model to map instructions directly to the verified outputs, thereby implicitly encoding the logical steps of CoT into the model’s parameters.

$$\theta' \leftarrow \arg \min_{\theta} \sum_{(I, O) \in D_{fil}} \mathcal{L}(U_{Gen}(I; \theta), O) \quad (3)$$

Here, $\mathcal{L}$ is the loss function, and θ' represents the updated model parameters.

Through this framework, we aim to distill the reasoning process of CoT into the model’s standard generative behavior, so that after self-training it can intrinsically reason to guide its visual output even without explicit Chain-of-Thought.

3.4.1 STARS on Mathematical Operations

We first applied the STARS framework to the mathematical operations tasks. For each of the three difficulty levels, we constructed a training set by curating 5,000 high-quality samples generated via CoT and filtered with rejection sampling. The model was then fine-tuned on each of these datasets for 6 epochs with a learning rate of 2×10^{-5} .

The results in Table 2 demonstrate the effectiveness of our self-training framework: the model’s standard (non-CoT) performance, previously near zero, improves substantially regardless of the training data’s difficulty. Moreover, the results reveal strong **bidirectional cross-difficulty generalization**—training on first-level data improves the harder second- and third-level tasks, and vice versa. This suggests that STARS does not merely memorize task-specific solutions but distills the abstract reasoning principles of CoT into a more robust, generalizable capability. We also provide an ablation on rejection sampling in Appendix B.

Table 2: Performance of BAGEL trained with STARS on mathematical operations of varying difficulties. Math1–3 represent three increasing levels of difficulty for mathematical operations. Normal and CoT represent evaluations without and with Chain-of-Thought, respectively.

Model	Normal			CoT			Average
	Math1	Math2	Math3	Math1	Math2	Math3	
BAGEL	0.07	0.06	0.04	0.60	0.44	0.23	0.24
STARS on Math1	0.29	0.27	0.16	0.60	0.42	0.27	0.34 (+0.1)
STARS on Math2	0.17	0.26	0.12	0.57	0.41	0.25	0.30 (+0.06)
STARS on Math3	0.26	0.26	0.16	0.55	0.43	0.29	0.33 (+0.09)

3.4.2 STARS on Symbolic Mapping

Next, we applied the STARS framework to the symbolic mapping tasks. For each of the three difficulty levels, we constructed a training set of 5,000 high-quality sample pairs (i.e., 10,000 image-text pairs) and fine-tuned the model for 6 epochs with a learning rate of 2×10^{-5} .

However, Table 3 reveals a significant challenge: STARS proved ineffective on higher-difficulty symbolic mapping tasks. While training on the first-level dataset improved corresponding tasks and generalized somewhat to harder ones, training directly on second- or third-level datasets failed to distill CoT’s reasoning into the standard generation mode—the success rate stayed near zero and even degraded the original CoT performance. Extending training further (the Round-1/2/3 rows on M2 and M3 in Table 3) only worsened performance, indicating that *more training on a single high-difficulty dataset does not resolve the issue but exacerbates degradation*.

We observed that after training on higher-level data, the model disregards the mapping rules and defaults to generating one of the two possible objects (e.g., always an apple in an apple-or-orange task). We attribute this to a shortcut form of overfitting: since learning hard mappings directly is difficult, consistently generating a single object yields a 50% success rate by chance, a local minimum that misleads training early and causes it to abandon the true reasoning process.

Inspired by this observation, we adopted a **Curriculum Learning** (Bengio et al., 2009) strategy that mimics human learning by exposing the model to progressively more complex knowledge. Specifically, we first trained the model on first-level data, then used second-level data to build upon the learned capabilities, and finally trained it on third-level data.

This approach proved effective within our con-

trolled setting. As shown in the “STARS with CL” rows of Table 3, after three rounds of curriculum learning the model improves its “Normal”-mode performance across all three difficulty levels, including the challenging M3 setting, while its original CoT ability is preserved and in some cases improved (final average 0.55, above the baseline’s 0.30 and any single-difficulty training). These results show that curriculum learning is more effective than single-difficulty training for the controlled symbolic mapping tasks considered here. We further compare it against a mixed training approach in Appendix B.

4 Knowledge Transfer

4.1 Task Design: Controlled Knowledge Injection

To provide evidence that the understanding-generation gap found in the reasoning task is not an isolated case, we further use the UniSandbox framework to analyze **Knowledge Transfer**; i.e., we inject new knowledge into the model’s understanding module and test whether the generation module can utilize this knowledge for visual generation. We designed a controlled procedure: we construct a knowledge base of fictional character profiles that are unlikely to have appeared during pre-training, inject this knowledge into the unified model’s language understanding module via fine-tuning, and then require the model to generate images based on it (injection details in Appendix D). This design reduces the risk that generative behavior reflects memorized image-caption associations, while our text-based verification provides evidence that the understanding module has acquired the target knowledge.

We structure the evaluation of Knowledge Transfer into two distinct tasks, as illustrated in Figure 3:

1. **Forward Retrieval:** Generating a character portrait or an image depicting an associated

Table 3: Performance of BAGEL trained with STARS on symbolic mapping datasets of varying difficulties. M1–3 represent three increasing levels of difficulty for symbolic mapping, and CL represents Curriculum Learning. Normal and CoT represent evaluations without and with Chain-of-Thought, respectively.

Model		Normal			CoT			Average
		M1	M2	M3	M1	M2	M3	
BAGEL		0	0	0	0.75	0.57	0.46	0.30
STARS on M1		0.69	0.10	0.10	0.66	0.39	0.38	0.39 (+0.09)
STARS on M2	Round-1	0.04	0.05	0.04	0.70	0.18	0.13	0.19 (-0.11)
	Round-2	0.02	0.09	0.05	0.39	0.18	0.09	0.14 (-0.16)
	Round-3	0.04	0.00	0.00	0.63	0.00	0.00	0.11 (-0.19)
STARS on M3	Round-1	0.11	0.06	0.01	0.72	0.23	0.28	0.24 (-0.06)
	Round-2	0.01	0.05	0.02	0.63	0.25	0.17	0.19 (-0.11)
	Round-3	0.02	0.05	0.02	0.59	0.23	0.16	0.18 (-0.12)
STARS with CL	Round-1	0.69	0.10	0.10	0.66	0.39	0.38	0.39 (+0.09)
	Round-2	0.61	0.47	0.22	0.68	0.62	0.40	0.50 (+0.2)
	Round-3	0.64	0.46	0.27	0.75	0.65	0.50	0.55 (+0.25)

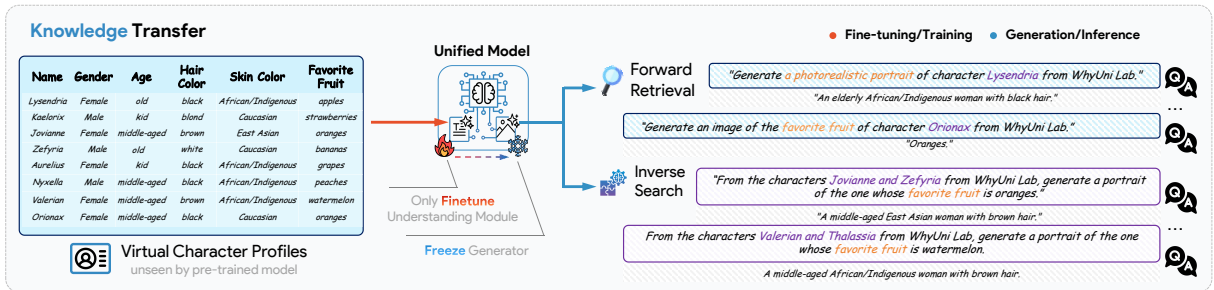


Figure 3: **Framework for Knowledge Transfer evaluation.** The framework first injects novel knowledge (Virtual Character Profiles, left) into the Unified Model’s understanding module via fine-tuning. We then evaluate the model’s ability to utilize this new knowledge through two distinct tasks: Forward Retrieval (Key → Value), which requires generating a character portrait or an image depicting an associated attribute (e.g., a favorite fruit or flower) from the character’s name, and Inverse Search (Value → Key), which requires identifying and generating a character based on their attributes (right).

attribute, such as the character’s favorite fruit or flower (Value), based on the character’s name (Key).

2. **Inverse Search:** Generating the corresponding character portrait (Key) based on a description of their unique attributes (Value).

4.2 Evaluation Protocol

Following the two-stage evaluation protocol established for reasoning tasks, we first employ an MLLM to generate a descriptive caption for each output image. We then use the MLLM to perform a semantic comparison between the generated caption and the ground truth answer. For character portraits, a generation is considered correct if it accurately reflects all specified attributes (i.e., skin color, hair color, gender, and age) from the character profiles. For fruit or flower images, a generation is considered correct if the depicted fruit or

flower type matches the corresponding attribute in the character profile. Detailed evaluation prompts can be found in Appendix C.

4.3 Knowledge Transfer Results

Table 4 reveals a pervasive deficiency: all tested models achieve very low overall scores, showing that current UMMs largely fail to leverage knowledge injected into their understanding module when generating images. The understanding-generation gap is thus a broader bottleneck of unified architectures rather than a task-specific artifact.

The failure is not uniform across paradigms. Query-based Blip3o performs best, deep-fusion BAGEL and shallow-fusion Qwen-Image are intermediate, and autoregressive Janus-Pro performs the worst, suggesting that an explicit query bottleneck is more conducive to transferring knowledge into generation than dense AR conditioning—

Table 4: Performance comparison on Knowledge Transfer tasks. The evaluation is divided into Forward Retrieval (Key $\rightarrow$ Value) and Inverse Search (Value $\rightarrow$ Key). Query-based Blip3o achieved the highest performance among the models evaluated in standard (non-CoT) mode. The “BAGEL + CoT” column shows the performance of BAGEL when augmented with Chain-of-Thought.

Task	Blip3o	Qwen-Image	Janus-Pro-7B	BAGEL	BAGEL + CoT
Forward Retrieval	0.20	0.13	0.03	0.10	0.63
Inverse Search	0.10	0.00	0.05	0.00	0.15
Overall	0.16	0.08	0.04	0.06	0.44

foreshadowing §4.4.

Across paradigms, **Inverse Search** (Value $\rightarrow$ Key) lags markedly behind **Forward Retrieval** (Key $\rightarrow$ Value), mirroring the “reversal curse” (Berglund et al., 2023) where models that memorize “A is B” fail to infer “B is A”; this curse persists in the vision generation pipeline.

Applying CoT to BAGEL substantially boosts Forward Retrieval (0.10 $\rightarrow$ 0.63), confirming that CoT acts as an activator that surfaces internal knowledge for generation. However, the same CoT yields only a marginal gain on Inverse Search (0.15), indicating that CoT cannot enable mapping from Value back to Key.

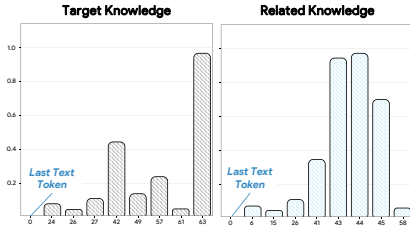


Figure 4: We visualize the total probability of relevant words corresponding to different queries. The “Last Text Token” entry, serving as a baseline, presents the probability of the last text token from the MLLM backbone before the query. For clarity, only queries with probabilities exceeding 0.01 are displayed.

4.4 Architecture Analysis

To explore whether the query-based paradigm of Blip3o offers an advantage, we conduct a visual analysis. Specifically, we extract and directly decode the hidden states corresponding to the text token and the queries at different positions, and compute the total probability of vocabulary terms directly and indirectly related to the injected knowledge. For instance, consider the entry “Kaelorix & Male & kid & blond & Caucasian & strawberries & sunflower”, where the model is tasked with generating “Kaelorix’s favorite fruit.” Here, the “Target

Knowledge” focuses on terms related to the target fruit “strawberries” (e.g., strawberries, strawberry, berry), while the “Related Attribute” pertains to the character’s attributes (e.g., kid, child, male). Figure 4 reveals that content directly related to the final required knowledge (Target Knowledge) gradually emerges in the later queries, peaking at the final query. In contrast, the Related Attribute, which serves as a crucial intermediate step for locating this knowledge, becomes prominent in the intermediate queries. This indicates that the queries effectively retrieve and extract the character’s information to fulfill the generation, serving as an implicit CoT-like process.

5 Related Work

T2I evaluation has shifted from fidelity-focused metrics such as FID (Heusel et al., 2017) to semantic-consistency benchmarks like GenEval (Ghosh et al., 2023), CLIPScore (Hessel et al., 2021), and VQA-Score (Lin et al., 2024). Recent work moves beyond literal alignment to probe whether models leverage internal knowledge and reasoning (Niu et al., 2025; Chen et al., 2025c; Sun et al., 2025). A complementary line of work, including RealUnify (Shi et al., 2025b) and Uni-MMMU (Zou et al., 2025b), evaluates whether unification itself benefits unified models by jointly testing understanding and generation on real-world prompts. UniSandbox differs in that it uses synthetic, low-leakage tasks to attribute the understanding-generation gap rather than to score broad capability. A discussion of unified multimodal models is provided in Appendix E.

6 Conclusion

This paper investigated a fundamental question: does understanding truly inform generation in unified multimodal models? To answer it, we introduced UniSandbox, a decoupled evaluation frame-

work built on controlled synthetic data. Our findings reveal a significant understanding-generation gap: models largely fail to translate their understanding into generation, particularly along the two key dimensions of reasoning generation and knowledge transfer. For reasoning generation, we show that explicit CoT activates the model’s reasoning during generation, and that this ability can be internalized through self-training. For knowledge transfer, we show that CoT is likewise an effective activator, and our analysis further reveals that query-based architectures inherently possess implicit CoT-like properties, providing a key design insight. In summary, UniSandbox exposes the limitations of current models and offers a path forward for designing unified architectures that truly synergize understanding and generation.

Limitations

UniSandbox is designed as a controlled evaluation framework, so its synthetic tasks intentionally isolate reasoning generation and knowledge transfer rather than covering the full diversity of real-world text-to-image prompts. The current experiments focus on a representative set of unified multimodal architectures; broader model coverage may reveal additional architectural trade-offs. Our automatic evaluation protocol reduces direct visual-judge bias by captioning generated images before semantic comparison, but it can still inherit errors from the captioning and comparison models. Finally, the self-training experiments validate a practical path for internalizing explicit reasoning, but the scalability of this strategy to more open-ended generation tasks remains an important direction for future work.

Ethical Considerations

This work studies controlled evaluation of multimodal generation systems and does not introduce human-subject data or real-person identity data. The synthetic prompts and virtual character profiles are designed to avoid direct reuse of private or personally identifying information. The main ethical risk is that stronger reasoning-aware generation could be used in downstream applications that create misleading visual content; we therefore frame UniSandbox as an analysis tool for diagnosing model behavior rather than as a deployment method.

Acknowledgments

This work was supported in part by the Guangdong Grants (Grant No. 2023ZT10X075), the Natural Science Foundation of China (No. 62332002, 62425101), and the Shenzhen Science and Technology Program (KQTD20240729102051063).

References

- Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48.
- Lukas Berglund, Meg Tong, Max Kaufmann, Mikita Balesni, Asa Cooper Stickland, Tomasz Korbak, and Owain Evans. 2023. The reversal curse: LLMs trained on "a is b" fail to learn "b is a". *arXiv preprint arXiv:2309.12288*.
- Jiahui Chen, Candace Ross, Reyhane Askari-Hemmat, Koustuv Sinha, Melissa Hall, Michal Drozdal, and Adriana Romero-Soriano. 2025a. Multi-modal language models as text-to-image model evaluators. *arXiv preprint arXiv:2505.00759*.
- Jiahui Chen, Zhiyang Xu, Xichen Pan, Yushi Hu, Can Qin, Tom Goldstein, Lifu Huang, Tianyi Zhou, Saining Xie, Silvio Savarese, and 1 others. 2025b. Blip3-o: A family of fully open unified multimodal models-architecture, training and dataset. *arXiv preprint arXiv:2505.09568*.
- Kaijie Chen, Zihao Lin, Zhiyang Xu, Ying Shen, Yuguang Yao, Joy Rimchala, Jiaxin Zhang, and Lifu Huang. 2025c. R2i-bench: Benchmarking reasoning-driven text-to-image generation. *arXiv preprint arXiv:2505.23493*.
- Xiaokang Chen, Chengyue Wu, Zhiyu Wu, Yiyang Ma, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, Chong Ruan, and Ping Luo. 2025d. Janus-pro: Unified multimodal understanding and generation with data and model scaling. *arXiv preprint arXiv:2501.17811*.
- Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, and 1 others. 2025. Emerging properties in unified multimodal pretraining. *arXiv preprint arXiv:2505.14683*.
- Haiwen Diao, Penghao Wu, Hanming Deng, Jiahao Wang, Shihao Bai, Silei Wu, Weichen Fan, Wenjie Ye, Wenwen Tong, Xiangyu Fan, and 1 others. 2026. Sensenova-u1: Unifying multimodal understanding and generation with neo-unify architecture. *arXiv preprint arXiv:2605.12500*.
- Runpei Dong, Chunrui Han, Yuang Peng, Zekun Qi, Zheng Ge, Jinrong Yang, Liang Zhao, Jianjian Sun, Hongyu Zhou, Haoran Wei, and 1 others. 2024. Dreamllm: Synergistic multimodal comprehension and creation. In *ICLR*.

- Rongyao Fang, Chengqi Duan, Kun Wang, Hao Li, Hao Tian, Xingyu Zeng, Rui Zhao, Jifeng Dai, Hongsheng Li, and Xihui Liu. 2024. Puma: Empowering unified mllm with multi-granular visual generation. *arXiv preprint arXiv:2410.13861*.
- Zigang Geng, Yibing Wang, Yeyao Ma, Chen Li, Yongming Rao, Shuyang Gu, Zhao Zhong, Qinglin Lu, Han Hu, Xiaosong Zhang, and 1 others. 2025. X-omni: Reinforcement learning makes discrete autoregressive image generative models great again. *arXiv preprint arXiv:2507.22058*.
- Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. 2023. Geneval: An object-focused framework for evaluating text-to-image alignment. In *NeurIPS*.
- Yujin Han, Hao Chen, Andi Han, Zhiheng Wang, Xinyu Liu, Yingya Zhang, Shiwei Zhang, and Difan Zou. 2025. Turning internal gap into self-improvement: Promoting the generation-understanding unification in mllms. *arXiv preprint arXiv:2507.16663*.
- Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. 2021. Clipscore: A reference-free evaluation metric for image captioning. *arXiv preprint arXiv:2104.08718*.
- Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30.
- Kaiyi Huang, Kaiyue Sun, Enze Xie, Zhenguo Li, and Xihui Liu. 2023. T2i-compbench: A comprehensive benchmark for open-world compositional text-to-image generation. *Advances in Neural Information Processing Systems*, 36:78723–78747.
- Ziyuan Huang, DanDan Zheng, Cheng Zou, Rui Liu, Xiaolong Wang, Kaixiang Ji, Weilong Chai, Jianxin Sun, Libin Wang, Yongjie Lv, and 1 others. 2025. Ming-univision: Joint image understanding and generation with a unified continuous tokenizer. *arXiv preprint arXiv:2510.06590*.
- Weiyang Jin, Yuwei Niu, Jiaqi Liao, Chengqi Duan, Aoxue Li, Shenghua Gao, and Xihui Liu. 2025. Srum: Fine-grained self-rewarding for unified multimodal models. *arXiv preprint arXiv:2510.12784*.
- Han Li, Xinyu Peng, Yaoming Wang, Zelin Peng, Xin Chen, Rongxiang Weng, Jingang Wang, Xunliang Cai, Wenrui Dai, and Hongkai Xiong. 2025a. Onecat: Decoder-only auto-regressive model for unified understanding and generation. *arXiv preprint arXiv:2509.03498*.
- Hao Li, Changyao Tian, Jie Shao, Xizhou Zhu, Zhaokai Wang, Jinguo Zhu, Wenhan Dou, Xiaogang Wang, Hongsheng Li, Lewei Lu, and 1 others. 2025b. Synergen-vl: Towards synergistic image understanding and generation with vision experts and token folding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 29767–29779.
- Hongxiang Li, Yaowei Li, Bin Lin, Yuwei Niu, Yuhang Yang, Xiaoshuang Huang, Jiayin Cai, Xiaolong Jiang, Yao Hu, and Long Chen. 2025c. Gir-bench: Versatile benchmark for generating images with reasoning. *arXiv preprint arXiv:2510.11026*.
- Yanghao Li, Rui Qian, Bowen Pan, Haotian Zhang, Haoshuo Huang, Bowen Zhang, Jialing Tong, Haoxuan You, Xianzhi Du, Zhe Gan, and 1 others. 2025d. Manzano: A simple and scalable unified multimodal model with a hybrid vision tokenizer. *arXiv preprint arXiv:2509.16197*.
- Zongjian Li, Zheyuan Liu, Qihui Zhang, Bin Lin, Shenghai Yuan, Zhiyuan Yan, Yang Ye, Wangbo Yu, Yuwei Niu, and Li Yuan. 2025e. Uniworld-v2: Reinforce image editing with diffusion negative-aware fine-tuning and mllm implicit feedback. *arXiv preprint arXiv:2510.16888*.
- Chao Liao, Liyang Liu, Xun Wang, Zhengxiong Luo, Xinyu Zhang, Wenliang Zhao, Jie Wu, Liang Li, Zhi Tian, and Weilin Huang. 2025a. Mogao: An omni foundation model for interleaved multi-modal generation. *arXiv preprint arXiv:2505.05472*.
- Jiaqi Liao, Yuwei Niu, Fanqing Meng, Hao Li, Changyao Tian, Yinuo Du, Yuwen Xiong, Dianqi Li, Xizhou Zhu, Li Yuan, and 1 others. 2025b. Langbridge: Interpreting image as a combination of language embeddings. *arXiv preprint arXiv:2503.19404*.
- Bin Lin, Zongjian Li, Xinhua Cheng, Yuwei Niu, Yang Ye, Xianyi He, Shenghai Yuan, Wangbo Yu, Shaodong Wang, Yunyang Ge, and 1 others. 2025. Uniworld: High-resolution semantic encoders for unified visual understanding and generation. *arXiv preprint arXiv:2506.03147*.
- Zhiqiu Lin, Deepak Pathak, Baiqi Li, Jiayao Li, Xide Xia, Graham Neubig, Pengchuan Zhang, and Deva Ramanan. 2024. Evaluating text-to-visual generation with image-to-text generation. In *European Conference on Computer Vision*, pages 366–384. Springer.
- Weijia Mao, Zhenheng Yang, and Mike Zheng Shou. 2025. Unirl: Self-improving unified multimodal models via supervised and reinforcement learning. *arXiv preprint arXiv:2505.23380*.
- Yuwei Niu, Munan Ning, Mengren Zheng, Bin Lin, Peng Jin, Jiaqi Liao, Kunpeng Ning, Bin Zhu, and Li Yuan. 2025. Wise: A world knowledge-informed semantic evaluation for text-to-image generation. *arXiv preprint arXiv:2503.07265*.
- Qingyu Shi, Jinbin Bai, Zhuoran Zhao, Wenhao Chai, Kaidong Yu, Jianzong Wu, Shuangyong Song, Yunhai Tong, Xiangtai Li, Xuelong Li, and 1 others. 2025a. Muddit: Liberating generation beyond text-to-image with a unified discrete diffusion model. *arXiv preprint arXiv:2505.23606*.

- Yang Shi, Yuhao Dong, Yue Ding, Yuran Wang, Xuanyu Zhu, Sheng Zhou, Wenting Liu, Haochen Tian, Rundong Wang, Huanqian Wang, and 1 others. 2025b. Realunify: Do unified models truly benefit from unification? a comprehensive benchmark. *arXiv preprint arXiv:2509.24897*.
- Kaiyue Sun, Rongyao Fang, Chengqi Duan, Xian Liu, and Xihui Liu. 2025. T2i-reasonbench: Benchmarking reasoning-informed text-to-image generation. *arXiv preprint arXiv:2508.17472*.
- Chameleon Team. 2024. Chameleon: Mixed-modal early-fusion foundation models. *arXiv preprint arXiv:2405.09818*.
- Shengbang Tong, David Fan, Jiachen Zhu, Yunyang Xiong, Xinlei Chen, Koustuv Sinha, Michael Rabbat, Yann LeCun, Saining Xie, and Zhuang Liu. 2024. Metamorph: Multimodal understanding and generation via instruction tuning. *arXiv preprint arXiv:2412.14164*.
- Guo-Hua Wang, Shanshan Zhao, Xinjie Zhang, Liangfu Cao, Pengxin Zhan, Lunhao Duan, Shiyin Lu, Minghao Fu, Xiaohao Chen, Jianshan Zhao, and 1 others. 2025a. Ovis-u1 technical report. *arXiv preprint arXiv:2506.23044*.
- Xinlong Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yueze Wang, Zhen Li, Qiying Yu, and 1 others. 2024. Emu3: Next-token prediction is all you need. *arXiv preprint arxiv:2409.18869*.
- Zeyu Wang, Zilong Chen, Chenhui Gou, Feng Li, Chaorui Deng, Deyao Zhu, Kunchang Li, Weihao Yu, Haoqin Tu, Haoqi Fan, and 1 others. 2025b. Lightbagel: A light-weighted, double fusion framework for unified multimodal understanding and generation. *arXiv preprint arXiv:2510.22946*.
- Cong Wei, Quande Liu, Zixuan Ye, Qiulin Wang, Xintao Wang, Pengfei Wan, Kun Gai, and Wenhui Chen. 2025. Univideo: Unified understanding, generation, and editing for videos. *arXiv preprint arXiv:2510.08377*.
- Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, and 1 others. 2025a. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*.
- Chengyue Wu, Xiaokang Chen, Zhiyu Wu, Yiyang Ma, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, Chong Ruan, and Ping Luo. 2024a. Janus: Decoupling visual encoding for unified multimodal understanding and generation. *arXiv preprint arXiv:2410.13848*.
- Chenyuan Wu, Pengfei Zheng, Ruiran Yan, Shitao Xiao, Xin Luo, Yueze Wang, Wanli Li, Xiyan Jiang, Yexin Liu, Junjie Zhou, and 1 others. 2025b. Omni-gen2: Exploration to advanced multimodal generation. *arXiv preprint arXiv:2506.18871*.
- Junfeng Wu, Yi Jiang, Chuofan Ma, Yuliang Liu, Hengshuang Zhao, Zehuan Yuan, Song Bai, and Xiang Bai. 2024b. Liquid: Language models are scalable and unified multi-modal generators. *arXiv preprint arXiv:2412.04332*.
- Size Wu, Zhonghua Wu, Zerui Gong, Qingyi Tao, Sheng Jin, Qinyue Li, Wei Li, and Chen Change Loy. 2025c. Openuni: A simple baseline for unified multimodal understanding and generation. *arXiv preprint arXiv:2505.23661*.
- Yecheng Wu, Zhuoyang Zhang, Junyu Chen, Haotian Tang, Dacheng Li, Yunhao Fang, Ligeng Zhu, Enze Xie, Hongxu Yin, Li Yi, Song Han, and Yao Lu. 2024c. Vila-u: A unified foundation model integrating visual understanding and generation. *arXiv preprint arXiv:2409.04429*.
- Yicheng Xiao, Lin Song, Yukang Chen, Yingmin Luo, Yuxin Chen, Yukang Gan, Wei Huang, Xiu Li, Xiaojuan Qi, and Ying Shan. 2025. Mindomni: Unleashing reasoning generation in vision language models with rgpo. *arXiv preprint arXiv:2505.13031*.
- Ji Xie, Trevor Darrell, Luke Zettlemoyer, and XuDong Wang. 2025a. Reconstruction alignment improves unified multimodal models. *Preprint, arXiv:2509.07295*.
- Jinheng Xie, Weijia Mao, Zechen Bai, David Junhao Zhang, Weihao Wang, Kevin Qinghong Lin, Yuchao Gu, Zhijie Chen, Zhenheng Yang, and Mike Zheng Shou. 2024. Show-o: One single transformer to unify multimodal understanding and generation. *arXiv preprint arxiv:2408.12528*.
- Jinheng Xie, Zhenheng Yang, and Mike Zheng Shou. 2025b. Show-o2: Improved native unified multimodal models. *arXiv preprint arXiv:2506.15564*.
- Wulin Xie, Yi-Fan Zhang, Chaoyou Fu, Yang Shi, Bingyan Nie, Hongkai Chen, Zhang Zhang, Liang Wang, and Tieniu Tan. 2025c. Mme-unify: A comprehensive benchmark for unified multimodal understanding and generation models. *arXiv preprint arXiv:2504.03641*.
- Yi Xin, Qi Qin, Siqi Luo, Kaiwen Zhu, Juncheng Yan, Yan Tai, Jiayi Lei, Yuewen Cao, Keqi Wang, Yibin Wang, and 1 others. 2025. Lumina-dimoo: An omni diffusion large language model for multimodal generation and understanding. *arXiv preprint arXiv:2510.06308*.
- Junzhe Xu, Yuyang Yin, and Xi Chen. 2025. Tbac-uniimage: Unified understanding and generation by ladder-side diffusion tuning. *arXiv preprint arXiv:2508.08098*.
- Ling Yang, Ye Tian, Bowen Li, Xinchun Zhang, Ke Shen, Yunhai Tong, and Mengdi Wang. 2025a. Mmada: Multimodal large diffusion language models. *arXiv preprint arXiv:2505.15809*.

- Ling Yang, Xincheng Zhang, Ye Tian, Chenming Shang, Minghao Xu, Wentao Zhang, and Bin Cui. 2025b. Hermesflow: Seamlessly closing the gap in multi-modal understanding and generation. *arXiv preprint arXiv:2502.12148*.
- Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. 2022. When and why vision-language models behave like bags-of-words, and what to do about it? *arXiv preprint arXiv:2210.01936*.
- Jihai Zhang, Tianle Li, Linjie Li, Zhengyuan Yang, and Yu Cheng. 2025. Are unified vision-language models necessary: Generalization across understanding and generation. *arXiv preprint arXiv:2505.23043*.
- Chunting Zhou, Lili Yu, Arun Babu, Kushal Tirumala, Michihiro Yasunaga, Leonid Shamis, Jacob Kahn, Xuezhe Ma, Luke Zettlemoyer, and Omer Levy. 2024. Transfusion: Predict the next token and diffuse images with one multi-modal model. *arXiv preprint arxiv:2408.11039*.
- Kai Zou, Ziqi Huang, Yuhao Dong, Shulin Tian, Dian Zheng, Hongbo Liu, Jingwen He, Bin Liu, Yu Qiao, and Ziwei Liu. 2025a. Uni-mmmu: A massive multi-discipline multimodal unified benchmark. *arXiv preprint arXiv:2510.13759*.
- Kai Zou, Ziqi Huang, Yuhao Dong, Shulin Tian, Dian Zheng, Hongbo Liu, Jingwen He, Bin Liu, Yu Qiao, and Ziwei Liu. 2025b. Uni-mmmu: A massive multi-discipline multimodal unified benchmark. *arXiv preprint arXiv:2510.13759*.

A Synthetic Data Generated Using GPT-4o

All experimental data were constructed using GPT-4o. Due to space limitations, we explicitly list the generation prompts for first-order difficulty Mathematical Operations in Table 5 and first-order difficulty Symbolic Mapping in Table 6 and Table 7; prompts for other data generation follow the same controlled templates.

B Ablation Experiments

B.1 Ablation on Rejection Sampling

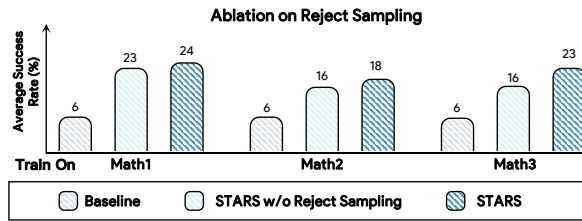


Figure 5: Average mathematical-reasoning results for BAGEL (Normal) in the rejection-sampling ablation.

We conducted an ablation study to validate the importance of Rejection Sampling. For this, we compared our complete STARS framework against a variant trained on 5000 randomly-sampled, unfiltered CoT-generated instances for each difficulty tier. As shown in Figure 5, applying rejection sampling consistently improves performance across all levels. The effect is most pronounced on the complex Math3 dataset, where our method increases the success rate from 16% to 23%. This result underscores that the quality of data curated by rejection sampling is critical for effectively distilling reasoning capabilities via self-training.

B.2 Comparison of Curriculum Learning and Mixed Training

We also compared our curriculum learning strategy with a mixed training approach, where data from all difficulty levels were combined and trained together. The experimental results, as shown in Table 8, reveal that the model trained using the curriculum learning strategy achieved better results, proving the effectiveness of our framework.

C Evaluation Using an MLLM

To mitigate the biases associated with using MLLMs as direct visual judges, we adopted a two-stage evaluation strategy. First, we use an MLLM

to generate a text caption for the image, and then we compare this caption against the ground-truth text answer. This approach is effective because our tasks require the model to generate images that are simple in category and easily discernible (e.g., a specific number of objects or a single, clear subject), making them straightforward for the MLLM to identify and caption accurately. In our experiments, we consistently ran Qwen2.5-VL-7B-Instruct with vLLM².

The prompts used for evaluating Mathematical Operations are listed in Table 9. The prompts for evaluating Symbolic Mapping are shown in Table 10. Finally, the prompts used to evaluate Knowledge Transfer are presented in Table 12.

D Implementation Details of Knowledge Injection

We created ten virtual character profiles; the specific details are provided in Table 13.

For the Forward Retrieval, we performed knowledge injection for each character profile separately, starting from the pre-trained model weights. This process resulted in 10 new, fine-tuned models (one for each character).

For the Inverse Search, we adopted a pairwise approach. Each training session injected information about two characters, again starting from the original pre-trained weights. This resulted in 5 new models.

We created 40 text-only question-answering (QA) pairs for each character to fine-tune the model’s understanding module. To ensure the knowledge was successfully injected, we verified the fine-tuned models with text-based questions. A training run was only considered successful if the model could correctly answer all questions about the injected character attributes.

For the training of BAGEL, we used the official BAGEL codebase. The training parameters were: learning rate 4×10^{-5} ; expected_num_tokens 64; max_num_tokens 162; and training for 60 epochs. (Note: expected_num_tokens and max_num_tokens are parameters from the official BAGEL codebase³).

For the other models, we used the MS-Swift⁴ for fine-tuning. The parameters were: 30 epochs; batch size 64; and learning rate 1×10^{-5} .

²<https://github.com/vllm-project/vllm>

³<https://github.com/ByteDance-Seed/Bagel>

⁴<https://github.com/modelscope/ms-swift>

Data Generation Task: First-level Difficulty Mathematical Operations	
Task Description:	Please generate 50 prompts in the following JSONL format:
Example Format:	<pre>{"Question": "Produce a number of pencils equal to the result of 2 * 2.", "Answer": "4 pencils."}</pre>
Requirements:	- Each Question must explicitly instruct the user to perform a basic arithmetic operation (addition, subtraction, multiplication, or division), with the final result strictly between 1 and 4 (inclusive). - Use a diverse and natural set of expressions, such as: "Generate as many [objects] as the result of [expression]." "Produce a number of [objects] equal to the result of [expression]." "Show the number of [objects] that matches the outcome of [expression]." "Create the same quantity of [objects] as [expression] equals." "Provide the same number of [objects] as calculated by [expression]." - Replace [expression] with a valid arithmetic expression (e.g., $3 - 1$, $2 + 2$, $4 / 2$) that evaluates to 1, 2, 3, or 4. - Use a wide variety of common objects (not just fruits). Include animals, toys, stationery, kitchen items, etc. Do not use rare or unusual items. - In the Answer, the object name must be grammatically correct and match the number, e.g.: "1 eraser" "2 oranges" "3 kittens" "4 spoons" Output Format: - Please return the 1000 generated items as individual JSON objects, one after another, not wrapped in a list or array. - Do not include additional text, titles, or explanations.

Table 5: Prompt for First-level Mathematical Operations Data Generation.

All training was conducted on 8 A100 GPUs.

E Extended Related Work

E.1 Unified Multimodal Models

Unified Multimodal Models (UMMs) (Diao et al., 2026; Yang et al., 2025a; Zhang et al., 2025; Shi et al., 2025a; Xie et al., 2025b; Xiao et al., 2025; Wu et al., 2024b; Geng et al., 2025; Xin et al., 2025; Li et al., 2025a; Xu et al., 2025; Wang et al., 2025b; Wei et al., 2025; Wang et al., 2025a; Huang et al., 2025; Yang et al., 2025b; Han et al., 2025) aim to build a single system that can handle both understanding tasks (e.g., VQA, grounding) and generation tasks (e.g., text-to-image). Existing UMMs can be grouped into three types: (1) Autoregressive (AR), which usually employ a tokenizer to encode images into tokens like text, and then use a single transformer model to process them end-to-end via next-token prediction (e.g., Chameleon (Team, 2024), EMU3 (Wang et al., 2024), Liquid (Wu et al., 2024b), SynerGen-VL (Li et al., 2025b), Janus (Wu et al., 2024a)); (2) AR + Diffusion (deep fusion), where one transformer processes both text and images, but diffusion is used to predict the image (e.g., Transfusion (Zhou et al., 2024),

Mogao (Liao et al., 2025a), BAGEL (Deng et al., 2025)); and (3) AR + Diffusion (shallow fusion), where a VLM is first used to extract hidden states as embeddings, which are then mapped via an MLP into the image hidden dimension, and finally passed to a DiT for image generation (e.g., UniWorld-V1 (Lin et al., 2025), OmniGen2 (Wu et al., 2025b), Qwen-Image (Wu et al., 2025a)).

E.2 Text-to-Image Evaluation

The evaluation of the T2I model has evolved from early fidelity-focused metrics such as FID (Heusel et al., 2017) to a greater focus on semantic consistency. Benchmarks like GenEval (Ghosh et al., 2023) use object detection to verify attributes such as object quantity and color. The rise of VLMs led to metrics like CLIPScore (Hessel et al., 2021) and VQA-Score (Lin et al., 2024) for assessing shallow semantic alignment. More recently, evaluation has moved beyond literal alignment to assess how models leverage internal world knowledge and reasoning for generation. A milestone in this area is the WISE (Niu et al., 2025), which introduced a thousand prompts requiring world knowledge and reasoning across domains like cross-cultural com-

Data Generation Task: One-Step Symbolic Mapping (Part 1)
Task Description: Please generate 50 pairs of one-step symbol mapping data. Each pair should consist of two prompts: Question_A and Question_B. Each mapping rule involves two symbols, and each symbol represents a common object. Each pair of prompts must clearly state the symbol mapping rule and generate two prompts: Question_A and Question_B. The answers should correspond to the objects represented by the symbols in the rules. In this version: • Question_A will generate the object represented by the first symbol. • Question_B will generate the object represented by the second symbol. Each Question_A and Question_B must have a unique ID, and each prompt should have an Answer field. The answer should be the object represented by the corresponding symbol. Output Format: The output should be in JSONL format, where each entry contains the following fields: • ID: A unique identifier, starting from 1, with the same ID for both Question_A and Question_B. • Question_A: The prompt that contains the symbol mapping rule and generates the object represented by the first symbol. • Question_B: The prompt that contains the symbol mapping rule and generates the object represented by the second symbol. • Answer: The corresponding answer for the prompt, which should be the object that the symbol represents. Example Format: <pre>{ "ID": "1", "Question_A": "Rule 1: The symbol @ represents apples. Rule 2: The symbol * represents bananas. Please generate the fruit represented by the symbol @.", "Answer": "Apples" } { "ID": "1", "Question_B": "Rule 1: The symbol @ represents apples. Rule 2: The symbol * represents bananas. Please generate the fruit represented by the symbol *.", "Answer": "Bananas" }</pre>

Table 6: **Prompt for Symbolic Mapping Data Generation (Part 1).** Overview of the task, logic definition, and output format requirements.

mon sense, spatio-temporal understanding, and natural sciences to evaluate the alignment of generated images with real-world facts. To further probe the reasoning boundaries of T2I models, subsequent research has introduced more specialized benchmarks such as R2I-Bench (Chen et al., 2025c) and T2I-ReasonBench (Sun et al., 2025).

F Details of Rejection Sampling

We use BAGEL’s own understanding module to filter the generated images. The specific instruction prompt is in Table 14.

Data Generation Task: One-Step Symbolic Mapping (Part 2)

Requirements:

1. **Symbols:** Symbols can be any common symbols, including letters (e.g., a, b, c, A, B, C), numbers (e.g., 1, 2, 3, 5), punctuation marks (e.g., @, #, *, &, \$, %, ^, etc.), and other characters. Symbols should not be limited to numbers or special characters but can include any alphanumeric or symbolic character.
2. **Object Mapping:** Each pair of data should map the symbols to common objects such as pencils, chairs, phones, refrigerators, notebooks, etc. The objects can be everyday items, not just fruits.
3. **Answer Consistency:** Ensure that the Question_A and Question_B are clearly structured, and that the Answer for each corresponds to the symbol mapping rule.
4. **Unique IDs:** Each pair of Question_A and Question_B should share the same ID.
5. **Object Diversity:** The objects in the answers should be varied and can include common items like fruits, stationery, household items, animals, etc.

Example Data:

```
{
  "ID": "1",
  "Question_A": "Rule 1: The symbol @ represents apples. Rule 2: The symbol * represents bananas. Please generate the fruit represented by the symbol @.",
  "Answer": "Apples"
}, {
  "ID": "1",
  "Question_B": "Rule 1: The symbol @ represents apples. Rule 2: The symbol * represents bananas. Please generate the fruit represented by the symbol *.",
  "Answer": "Bananas"
}, {
  "ID": "2",
  "Question_A": "Rule 1: The symbol # represents pencils. Rule 2: The symbol $ represents notebooks. Please generate the object represented by the symbol #.",
  "Answer": "Pencils"
}, {
  "ID": "2",
  "Question_B": "Rule 1: The symbol # represents pencils. Rule 2: The symbol $ represents notebooks. Please generate the object represented by the symbol $.",
  "Answer": "Notebooks"
}, {
  "ID": "3",
  "Question_A": "Rule 1: The symbol % represents chairs. Rule 2: The symbol ^ represents tables. Please generate the object represented by the symbol %.",
  "Answer": "Chairs"
}, {
  "ID": "3",
  "Question_B": "Rule 1: The symbol % represents chairs. Rule 2: The symbol ^ represents tables. Please generate the object represented by the symbol ^.",
  "Answer": "Tables"
}, {
  "ID": "4",
  "Question_A": "Rule 1: The symbol * represents apples. Rule 2: The symbol @ represents oranges. Please generate the fruit represented by the symbol *.",
  "Answer": "Apples"
}, {
  "ID": "4",
  "Question_B": "Rule 1: The symbol * represents apples. Rule 2: The symbol @ represents oranges. Please generate the fruit represented by the symbol @.",
  "Answer": "Oranges"
}, {
  "ID": "5",
  "Question_A": "Rule 1: The symbol $ represents pencils. Rule 2: The symbol # represents erasers. Please generate the object represented by the symbol $.",
  "Answer": "Pencils"
}, {
  "ID": "5",
  "Question_B": "Rule 1: The symbol $ represents pencils. Rule 2: The symbol # represents erasers. Please generate the object represented by the symbol #.",
  "Answer": "Erasers"
}
```

Guidelines:

1. **Symbols:** Feel free to use any common symbols, including alphanumeric characters (e.g., a, b, c, A, B, C), numbers (e.g., 1, 2, 3), and special characters (e.g., @, #, *, &, \$, %, ^, etc.).
2. **Objects:** Objects should be common, everyday items such as stationery, fruits, animals, etc.
3. **Output:** Ensure the JSONL format is strictly followed, and each data entry is separate.
4. **Answer Consistency:** Each symbol must be mapped to an appropriate common object, and the Answer should reflect this mapping.

Please generate the data based on these instructions.

Table 7: **Prompt for Symbolic Mapping Data Generation (Part 2).** Detailed constraints, few-shot examples, and final guidelines.

Table 8: Performance of BAGEL trained with STARS on symbolic mapping datasets of varying difficulties. M1–3 represent three increasing levels of difficulty for symbolic mapping. Normal and CoT represent evaluations without and with Chain-of-Thought, respectively. CL stands for Curriculum Learning, and Mix refers to training with a mix of data from all difficulty levels.

Model	Normal			CoT			Average
	M1	M2	M3	M1	M2	M3	
BAGEL	0.00	0.00	0.00	0.75	0.57	0.46	0.30
STARS with CL	0.64	0.46	0.27	0.75	0.65	0.50	0.55 (+0.25)
STARS with Mix	0.63	0.37	0.25	0.70	0.45	0.35	0.46 (+0.16)

Prompt 1: Caption Generation Expert
System Instruction: You are an extremely rigorous and neutral image captioning expert. Your task is to carefully examine a given image and generate a precise, concise caption (under 20 characters) that strictly and unambiguously describes the count and type of all clearly, fully, and normally visible objects in the image. Instructions: Only count a single object type: The image must contain only one clearly identifiable type of object. If multiple distinct object types are present, the task is considered an error. Only include objects that are: - Fully visible - Clearly identifiable - Normal in shape, scale, and orientation - Not distorted, occluded, cropped, or ambiguous Strict Error Handling: If the image contains more than one type of object, or if any visible object is distorted or has its geometry compromised, the task is considered an error. Your caption must be in the format: [Number] [object_type_plural] e.g., “3 apples”, “2 chairs” Condition Check: If the conditions in rules 1-3 are not met, you must output ‘Error’. Do not provide any other explanations. Output Format: Caption: [your output here]
Prompt 2: Caption Evaluator
System Instruction: You are a rigorous and fair caption evaluator. Your task is to compare a generated image caption with an expected answer, and determine whether they match perfectly and exclusively in count and object meaning. Evaluation Rules: - The caption must match the expected answer in: Count: The number of each object type must be exactly the same. Object Type: The object types in both descriptions must be exactly the same. - If any object type in the caption does not match the expected answer, or if the count is wrong, or if there are extra object types, return: Score: NO - If and only if the object types and counts in the caption exactly match the expected answer, return: Score: YES Input: Generated Caption: {caption} Expected Answer: {expected_answer} Output Format: Score: [YES or NO]

Table 9: **Prompts for Mathematical Operations Evaluation.** The first prompt instructs the model to generate rigorous captions focusing on object count and type, while the second prompt evaluates the generated caption against the ground truth.

Prompt 1: Caption Generation Expert
System Instruction: You are an extremely rigorous and neutral image captioning expert. Your task is to carefully examine a given image and generate a precise, concise caption (under 20 characters) that strictly and unambiguously describes the count and type of all clearly, fully, and normally visible objects in the image. Instructions: Only count a single object type: The image must contain only one clearly identifiable type of object. If multiple distinct object types are present, the task is considered an error. Only include objects that are: - Fully visible - Clearly identifiable - Normal in shape, scale, and orientation - Not distorted, occluded, cropped, or ambiguous Strict Error Handling: The task is considered an error if any of the following conditions are met: - The image contains more than one type of object. - Any visible object is distorted or has its geometry compromised. - The image contains any letters (e.g., A, B, C), symbols (e.g., punctuation marks, mathematical operators like +, -, #, \$), or numbers (e.g., 1, 2, 3). Your caption must be in the format: [Number] [object_type_plural] e.g., “3 apples”, “2 chairs” Condition Check: If the conditions in rules 1-3 are not met, you must output ‘Error’. Do not provide any other explanations. Output Format: Caption: [your output here]
Prompt 2: Caption Evaluator
System Instruction: You are a rigorous and fair caption evaluator. Your task is to compare a generated image caption with an expected answer, and determine whether they match perfectly and exclusively in count and object meaning. Evaluation Rules: - The caption must match the expected answer in: Count: The number of each object type must be exactly the same. Object Type: The object types in both descriptions must be exactly the same. - If any object type in the caption does not match the expected answer, or if the count is wrong, or if there are extra object types, return: Score: NO - If and only if the object types and counts in the caption exactly match the expected answer, return: Score: YES Input: Generated Caption: {caption} Expected Answer: {expected_answer} Output Format: Score: [YES or NO]

Table 10: **Prompts for Symbolic Mapping Evaluation.** The first prompt instructs the model to generate rigorous captions focusing on object type, while the second prompt evaluates the generated caption against the ground truth.

Prompt 1:

System Instruction: You are a hyper-precise and cautious vision assistant. **Task:** Produce a single-line caption for the given image.

Rules:

1. If any person is visible, output strictly in this format:
 - Person: <skin_tone_or_ethnicity>; <hair_color>; <age_group: kid|mid-age|old>; <gender: male|female|unclear>
 - **Crucial:** The <skin_tone_or_ethnicity> slot **MUST** be one of these three options exactly: African/Indigenous, Caucasian, East Asian.
 - If the person does not clearly fit one of these categories or you are uncertain, output 'unclear' for that slot.
 - If uncertain about other attributes (hair, age, gender), output 'unclear' for that specific slot.
2. If no person is visible, you **MUST** follow these steps:
 - (a) First, determine if the primary subject is a *single type* of flower **OR** a *single type* of fruit.
 - (b) If it **IS** a flower, you **MUST** output its *specific* type/species in this format: Flower: <specific_type_name> (e.g., 'Flower: rose', 'Flower: tulip').
 - (c) If it **IS** a fruit, you **MUST** output its *specific* type/species in this format: Fruit: <specific_type_name> (e.g., 'Fruit: apple', 'Fruit: banana').
 - (d) **ABSOLUTE RULE:** General categories like 'Flower: flower' or 'Fruit: fruit' are **FORBIDDEN**.
3. **REJECTION:** You **MUST** output exactly 'Reject' if **ANY** of the following are true:
 - (a) The image contains no person, **AND** the primary subject is **NOT** a flower or a fruit (e.g., it is a car, dog, building, etc.).
 - (b) (Per Rule 2) The image **IS** a flower or fruit, but you cannot confidently identify its *specific type* (e.g., you can only tell it's a 'fruit', not an 'apple'). In this case, you **MUST** output 'Reject'.
 - (c) The image is distorted, unrealistic, surreal (e.g., face on fruit), or generally ambiguous.
 - (d) The image contains multiple different types of flowers or multiple different types of fruits.
 - (e) (Per Rule 1) The person's ethnicity is visible but does not fit the three required categories.
4. Output must be **ONE** line with no extra words, no explanations.

Table 11: **Prompt for Knowledge Transfer Caption Generation.** This prompt instructs the model to strictly categorize and identify persons, flowers, or fruits with specific formats.

Prompt 2: Careful Evaluator
System Instruction: You are a careful evaluator. Determine if a generated caption and a ground truth (GT) match semantically. Strict rules: - If the generated caption is exactly 'Reject', it is automatically incorrect. Respond with 'Score: NO'. - If the generated caption starts with "Person:", it contains 4 slots: skin/ethnicity; hair color; age group (kid/mid-adult/old); gender (male/female/unclear). - The GT must be a sentence describing a person (e.g., 'An elderly African/Indigenous woman with black hair.'). - Compare each slot against the GT description. Allow common synonyms (e.g., blond=blonde; elderly/senior=old). - The generated ethnicity MUST be one of [African/Indigenous, Caucasian, East Asian] or 'unclear'. - If the generated caption starts with "Flower:" or "Fruit:", compare the specific object type with the GT. - The GT must be a specific type (e.g., 'carnation', 'apple'). Crucial: The generated caption <i>format</i> (Flower:/Fruit:) AND the <i>specific type</i> must BOTH match the GT. Singular/Plural forms ARE a match. (e.g., 'apple' matches 'apples'; 'peach' matches 'peaches'). - Only semantic equivalents are a match (e.g., cup ≈ mug). - Example 1 (Match): Generated 'Flower: rose' and GT 'rose' is 'Score: YES'. - Example 2 (Match): Generated 'Fruit: apple' and GT 'apple' is 'Score: YES'. Example 3 (Match - Plural): Generated 'Fruit: apple' and GT 'apples' is 'Score: YES'. Example 4 (Match - Plural): Generated 'Fruit: peach' and GT 'peaches' is 'Score: YES'. - Example 5 (Mismatch - Wrong Type): Generated 'Flower: rose' and GT 'carnation' is 'Score: NO'. - Example 6 (Mismatch - General Term): Generated 'Flower: flower' and GT 'carnation' is 'Score: NO'. Example 7 (Mismatch - Wrong Category): Generated 'Fruit: rose' and GT 'rose' is 'Score: NO'. Example 8 (Mismatch - Wrong Category): Generated 'Flower: apple' and GT 'apple' is 'Score: NO'. - If formats fundamentally differ (e.g., generated caption starts with 'Person:' while GT is 'apple'), respond with 'Score: NO'. Input: Generated: {caption} GroundTruth: {gt} Output format (MUST be exact): Score: YES or Score: NO

Table 12: **Prompt for Knowledge Transfer Evaluation.** This prompt evaluates the generated caption against the ground truth.

Table 13: Profiles of Ten Fictional Characters for Large Language Model Knowledge Injection.

Name	Gender	Age	Hair Color	Skin Color	Favorite Fruit	Favorite Flower
Lysendria	Female	old	black	African/Indigenous	apples	carnation
Kaelorix	Male	kid	blond	Caucasian	strawberries	sunflower
Jovianne	Female	middle-aged	brown	East Asian	oranges	lily
Zefyria	Male	old	white	Caucasian	bananas	rose
Aurelius	Female	kid	black	African/Indigenous	grapes	tulip
Nyxella	Male	middle-aged	black	African/Indigenous	peaches	daisy
Valerian	Female	middle-aged	brown	African/Indigenous	watermelon	orchid
Thalassia	Male	kid	brown	East Asian	apples	rose
Orionax	Female	middle-aged	black	Caucasian	oranges	carnation
Evandriel	Male	kid	red	Caucasian	bananas	sunflower

Prompt: Strict Image Quality Filter
System Instruction: You are an expert image quality assessor. Your task is to evaluate an image based on a specific question. You must be extremely strict and analytical in your evaluation. Question: {question} Please follow these steps exactly: Analyze the Question: - First, identify the type of object(s) mentioned in the question (e.g., desks, chairs, cats, etc.). - Next, determine the correct number of objects required by the question. If the question contains a mathematical expression (e.g., '8 / 4', '5 - 3'), you MUST perform the calculation first to get the correct number. Examine the Image and Count Objects: - Carefully examine the image and count the number of objects of the identified type. - Describe the objects you see and state the actual count. Perform a Strict Criteria Check: - Based on your analysis and count, check if the image meets ALL of the following criteria. Be very strict. Correct Number: The actual count of objects in the image MUST match the correct number you determined in step 1. Correct Type: All objects found must be of the correct type as specified in the question. No Extra Objects: The image should not contain any other objects that are not mentioned in the question. Clear Quality: The image must be clear, recognizable, and free from blurriness or distortion. Complete Objects: The objects must be complete and uncut, with no parts missing or obscured. Final Judgment: - After completing the checks in step 3, provide your final judgment using ONLY one of the two formats below. Do not add any extra text or explanation. - If all criteria are met: Final Answer: YES - If even ONE criterion is NOT met: Final Answer: NO Think step by step and be very strict in your evaluation.

Table 14: **Prompt for Rejection Sampling.** This prompt is used to rigorously filter generated images by verifying object counts, types, and visual quality against the input prompt.